Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

1. (Currently amended) A compiler for producing an object program from a source program ~~used~~ to be executed on an architecture equipped with a plurality of memory hierarchies ~~from a source program in conjunction with a computer system~~, the compiler configured to operate a computer to perform steps comprising:

a step for detecting ~~interpreting either an option or~~ a designation statement designating which memory hierarchy among the plurality of memory hierarchies will serves as the main data store for an object program ~~a target program mainly refers to data present in,~~ when the ~~target~~ object program is executed; [[and]]

a step for performing an optimizing process directed to said designated memory hierarchy to produce the object program; and

a step for storing the object program on a data store.

2. (Currently amended) A compiler as claimed in claim 1, wherein:

~~as said optimizing process directed to the designated memory hierarchy,~~ a memory latency is calculated according to the designated memory hierarchy with respect to an instruction for accessing a memory location in the designated memory hierarchy; ~~and an~~ wherein said optimizing process ~~responding to~~ is based on the calculated latency ~~is carried out~~.

3. (Currently amended) A compiler as claimed in claim 1, wherein:

~~as said optimizing process directed to the designated memory hierarchy,~~ a loop transformation method of a loop interchange, a loop unrolling, or a loop tiling is determined according to the designated memory hierarchy with respect to a memory access instruction; and wherein said optimizing process is based thereon.

4. (Currently amended) An object program producing method executed by both a computer system and a compiler executing on the computer system for producing an object program ~~used~~ to be executed on an architecture equipped with a plurality of memory hierarchies from a source program ~~in conjunction with a computer system~~,

said method comprising:

a step for detecting ~~interpreting either an option or~~ a designation statement designating which memory hierarchy ~~a target~~ an object program mainly refers to for storing and accessing data ~~present in,~~ when the ~~target~~ object program is executed; [[and]]

a step for performing an optimizing process directed to said designated memory hierarchy to produce the object program; and

a step for outputting and storing the object program on a data store.

5. (Currently amended) A code producing method as claimed in claim 4, wherein:

[[as]]for said optimizing process directed to the designated memory hierarchy, a memory latency is calculated according to the designated memory hierarchy with respect to a memory access instruction; and an optimizing process according to the calculated latency is carried out.

6. (Currently amended) A code producing method as claimed in claim 4, wherein:

[[as]]for said optimizing process directed to the designated memory hierarchy, a loop transformation method of a loop interchange, a loop unrolling, or a loop tiling is determined according to the designated memory hierarchy with respect to a memory access instruction.

7. (Canceled)

8. (Original) A storage medium wherein:

said storage medium has stored thereinto the compiler recited in claim 1.

9. (Currently amended) A method for producing an object program ~~used~~ to be executed on an architecture equipped with a plurality of memory hierarchies from a source program in conjunction with a computer system, wherein:

said computer system executes:

a step for ~~analyzing~~ detecting a designation statement designating which hierarchy an object program mainly refers to for storing data ~~stored in a memory of,~~ when said object program is executed; [[and]]

a step for producing said object program in which an optimizing process including different processes sequences according to said plural memory hierarchies is carried out with respect to said source program~~, and an object program which has been optimized as to an access to said memory hierarchy is produced~~ by selecting a process[[es]] sequence corresponding to the memory hierarchy designated by said designation statement; and

a step for storing the object program on a data storage device.

10. (Original) An object program producing method as claimed in claim 9, wherein:

said designation statement is described in an option within a compiler initiating command.

11. (Original) An object program producing method as claimed in claim 9, wherein:

said designation statement is inserted into said source program.

12. (Original) An object program producing method as claimed in claim 11, wherein:

said designation statement is applied to each of plural loops contained in said source program;

said analysis step includes a step for forming a loop table indicative of a correspondence relationship between the respective loops and the memory hierarchies designated by the designation statements corresponding to said loops; and

said execution step includes a step for acquiring a memory hierarchy designated by said designation statement by referring to said loop table.

13. (Original) An object program producing method as claimed in claim 9, wherein:

said memory hierarchies include a hierarchy constructed of a primary cache, a hierarchy constructed of a secondary cache, and a hierarchy constructed of a main storage apparatus.

14. (Original) An object program producing method as claimed in claim 9, wherein:

said optimizing process contains at least one of an optimizing process by instruction scheduling, a prefetch optimizing process, and an optimizing process by loop tiling and loop interchange/loop unrolling.

15. (Original) An object program producing method as claimed in claim 14, wherein:

said optimizing process corresponds to the optimizing process by the instruction scheduling; and a number of memory access latency cycles to be set are different from each other according to said memory hierarchies in said processes sequence.

16. (Original) An object program producing method as claimed in claim 14, wherein:

said optimizing process corresponds to the prefetch optimizing process; and timing of a prefetch code to be inserted is different from each other according to said memory hierarchies in said processes sequence.

17. (Original) An object program producing method as claimed in claim 14, wherein:

said optimizing process corresponds to the optimizing process by the loop tiling; a tile size is different from each other according to said memory hierarchies in said processes sequence.

18. (Original) An object program producing method as claimed in claim 14, wherein:

said optimizing process corresponds to the optimizing process by the loop interchange/loop unrolling; and in said processes sequence, it is determined to apply, or not to apply either the loop interchange or the loop unrolling according to said memory hierarchies.

19. (Currently amended) An apparatus for producing an object program used to be executed on an architecture equipped with a plurality of memory hierarchies from a source program, comprising:

a storage apparatus for previously storing thereinto an optimizing process containing different process[[es]] sequences according to said plurality of memory hierarchies;

an input apparatus for inputting said source program and a designation statement designating which memory hierarchy an object program mainly refers to data present in, when said object program is executed;

a processing apparatus for producing an optimized object program based upon both said source program and said designation statement; and

an output apparatus for outputting said optimized object program; wherein:

said processing apparatus executes:

a step for analyzing said designation statement;

a step for producing an object program which has been optimized as to an access to said memory hierarchy by selecting a processes sequence corresponding to the memory hierarchy designated by said designation statement; and

a step for outputting said optimized object program ~~form~~ from said output apparatus.